

Projekt zaliczeniowy

JAVA - Programowanie Sieciowe



Grupa L5

Piotr hadała

ARKADIUSZ NOWAK

PAWEŁ JĘDRZEJCZAK

NATALIA GÓROWSKA

Spis treści

[1. Wstęp 2](#_Toc157536533)

[2. BookApiController 2](#_Toc157536534)

[Adres do API 3](#_Toc157536535)

[Otrzymanie książki po ID 3](#_Toc157536536)

[Książki rekomendowane 4](#_Toc157536537)

[3. GenreApiController 5](#_Toc157536538)

[Adres do API 5](#_Toc157536539)

[Otrzymanie informacji o gatunku po jego nazwie 6](#_Toc157536540)

[5. Walidacja zapytań 7](#_Toc157536541)

[6. Baza danych 7](#_Toc157536542)

# Wstęp

Projekt robiony na ocenę bardzo dobry. W zadaniach na ocenę 3.0 zrobione polecenia z wariantu numer 1. W tym pliku opisana jest dokumentacja do API(jedno z poleceń na ocenę 3.0). Są tu utworzone dwa API, jedno do zwracania danych na temat książek znajdujących się w bazie danych. Drugie natomiast zwraca informacje odnośnie gatunków książek.

# BookApiController

To API pozwoli nam na otrzymywanie informacji na temat książek znajdujących się w bazie danych.

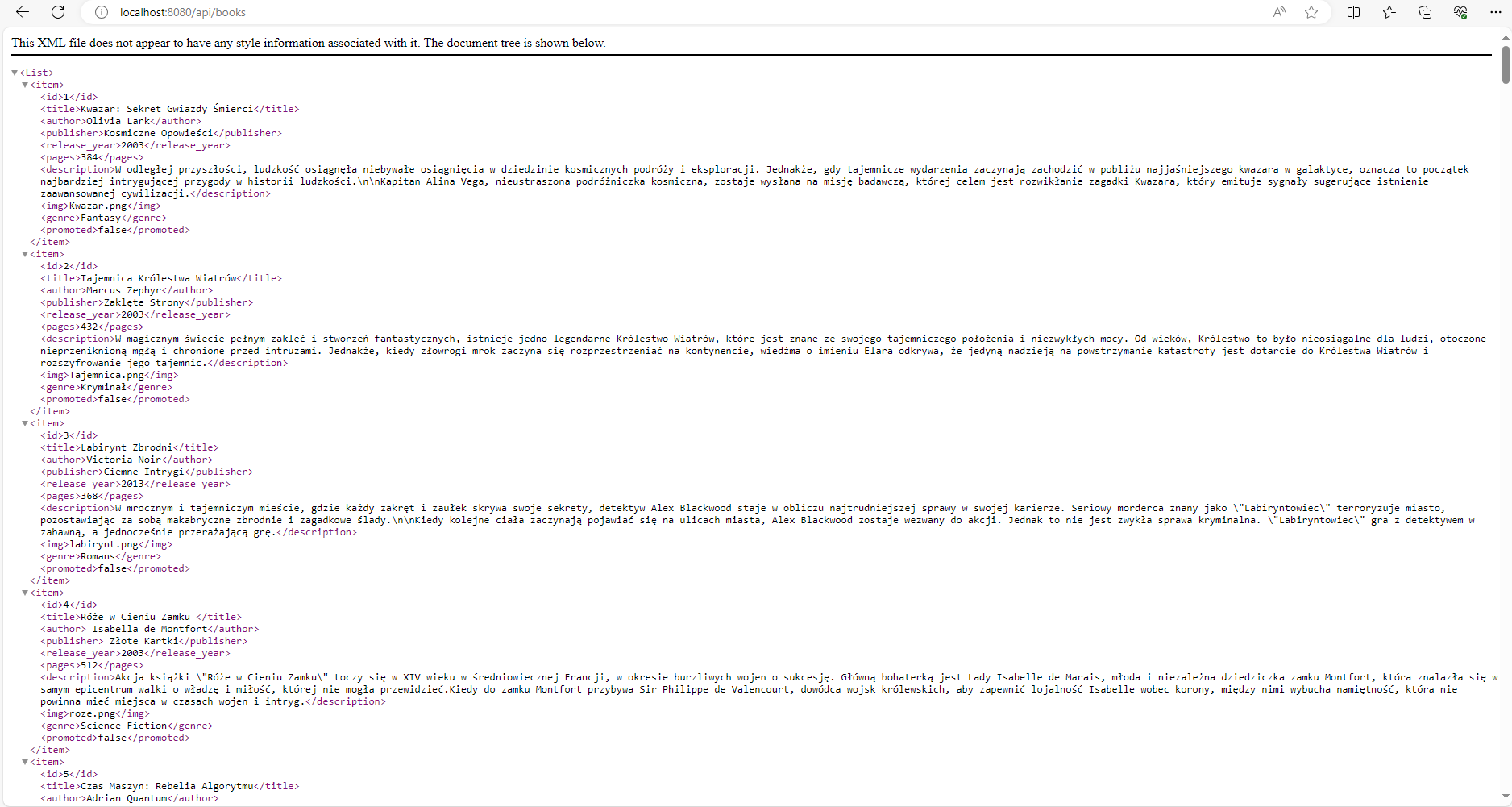
Plik BookApiController.java:

package pl.library.libraryonline.web.api;  
  
import org.springframework.http.HttpStatus;  
import org.springframework.http.MediaType;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.web.server.ResponseStatusException;  
import pl.library.libraryonline.domain.book.BookService;  
import pl.library.libraryonline.domain.book.dto.BookDto;  
  
import java.util.List;  
  
@RestController  
@RequestMapping(  
 path = "/api/books",  
 produces = {  
 MediaType.*APPLICATION\_JSON\_VALUE*,  
 MediaType.*APPLICATION\_XML\_VALUE* })  
public class BookApiController {  
 private final BookService bookService;  
  
 public BookApiController(BookService bookService) {  
 this.bookService = bookService;  
 }  
  
 @GetMapping()  
 public List<BookDto> getAllBooks() {  
 return bookService.findAllBooks();  
 }  
  
 @GetMapping("/{id}")  
 public BookDto getSingleBook(@PathVariable long id) {  
 BookDto book = bookService.findBookById(id)  
 .orElseThrow(() -> new ResponseStatusException(HttpStatus.*NOT\_FOUND*));  
 return book;  
 }  
 @GetMapping("/recommended")  
 public List<BookDto> getRecommendedBooks() {  
 return bookService.findAllPromotedBooks();  
 }  
}

## Adres do API

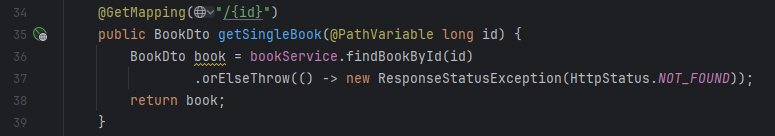
Aby otrzymać dane z API należy dodać do adresu ***/api/books*.** Otrzymamy wtedy listę wszystkich pozycji znajdujących się w bazie.

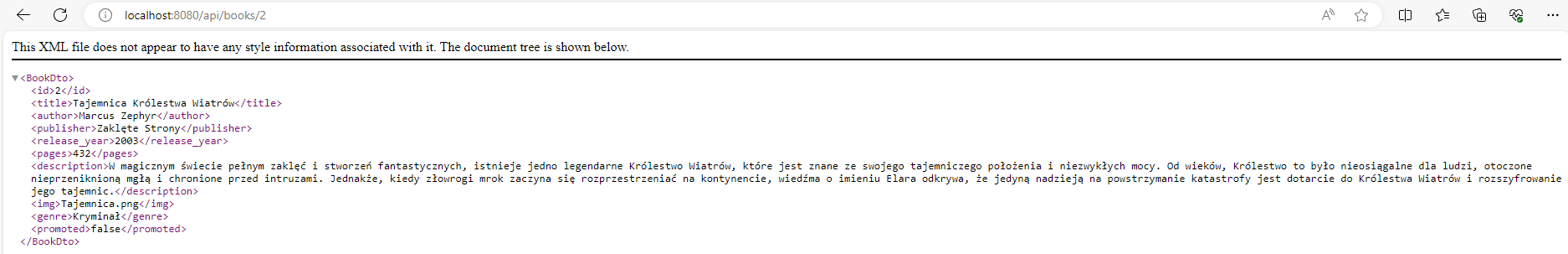




## Otrzymanie książki po ID

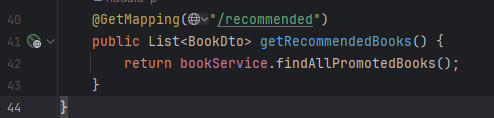
Po wpisaniu /api/books/{id}, np. **/api/books/2** otrzymamy informacje o pozycji z ID równym 2.





## Książki rekomendowane

Ścieżka **/api/books/recommended** zwróci nam pozycje, dla których wartość recommended wynosi true.





# GenreApiController

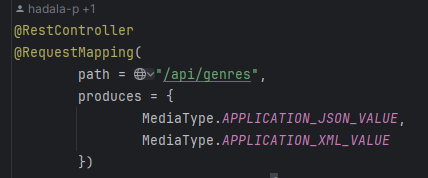
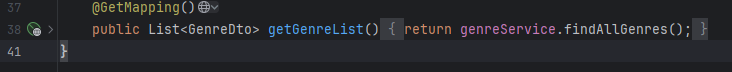
To API pozwoli nam na otrzymywanie informacji na gatunków książek które posiadamy w naszej bazie danych.

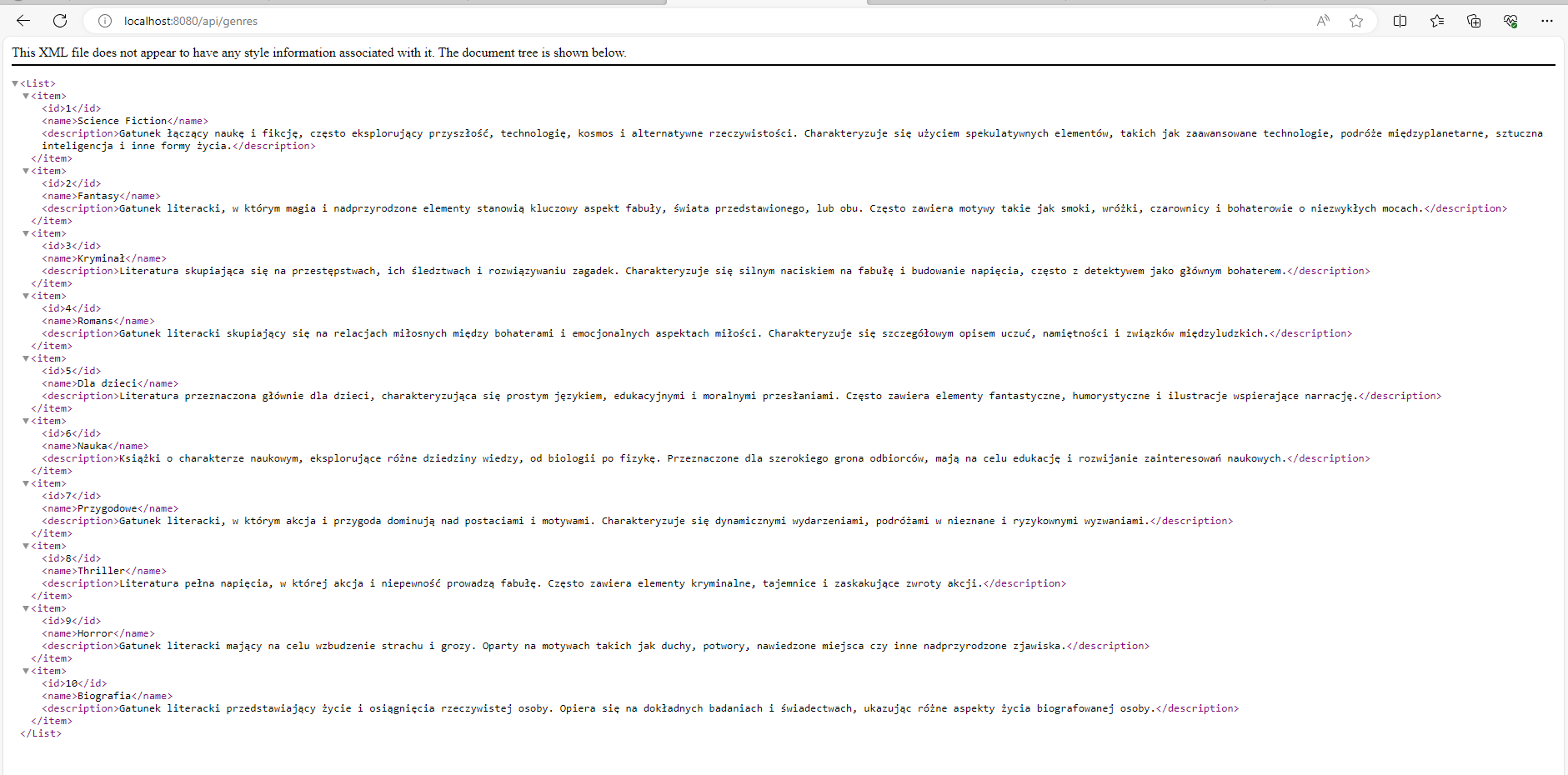
Plik GenreApiController.java:

1. package pl.library.libraryonline.web.api;  
     
   import org.springframework.http.HttpStatus;  
   import org.springframework.http.MediaType;  
   import org.springframework.web.bind.annotation.GetMapping;  
   import org.springframework.web.bind.annotation.PathVariable;  
   import org.springframework.web.bind.annotation.RequestMapping;  
   import org.springframework.web.bind.annotation.RestController;  
   import org.springframework.web.server.ResponseStatusException;  
   import pl.library.libraryonline.domain.book.BookService;  
   import pl.library.libraryonline.domain.genre.GenreService;  
   import pl.library.libraryonline.domain.genre.dto.GenreDto;  
     
   import java.util.List;  
     
   @RestController  
   @RequestMapping(  
    path = "/api/genres",  
    produces = {  
    MediaType.*APPLICATION\_JSON\_VALUE*,  
    MediaType.*APPLICATION\_XML\_VALUE* })  
   public class GenreApiController {  
    private final GenreService genreService;  
     
    public GenreApiController(GenreService genreService) {  
    this.genreService = genreService;  
    }  
     
    @GetMapping("/{name}")  
    public GenreDto getGenre(@PathVariable String name) {  
    GenreDto genre = genreService.findGenreByName(name)  
    .orElseThrow(() -> new ResponseStatusException(HttpStatus.*NOT\_FOUND*));  
    return genre;  
    }  
     
    @GetMapping()  
    public List<GenreDto> getGenreList() {  
    return genreService.findAllGenres();  
    }  
   }

## Adres do API

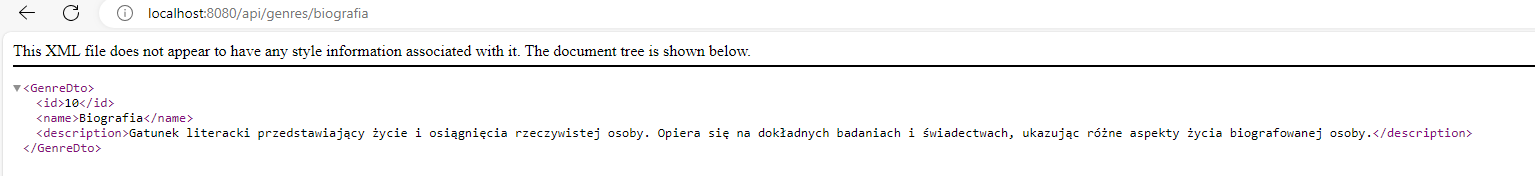
Aby otrzymać dane z API należy dodać do adresu ***/api/genres*.** Otrzymamy wtedy listę wszystkich pozycji znajdujących się w bazie.





## Otrzymanie informacji o gatunku po jego nazwie

Po wpisaniu /api/genres/{name}, np. **/api/genres/biografia** otrzymamy informacje związane z rekordem o nazwie Biografia.

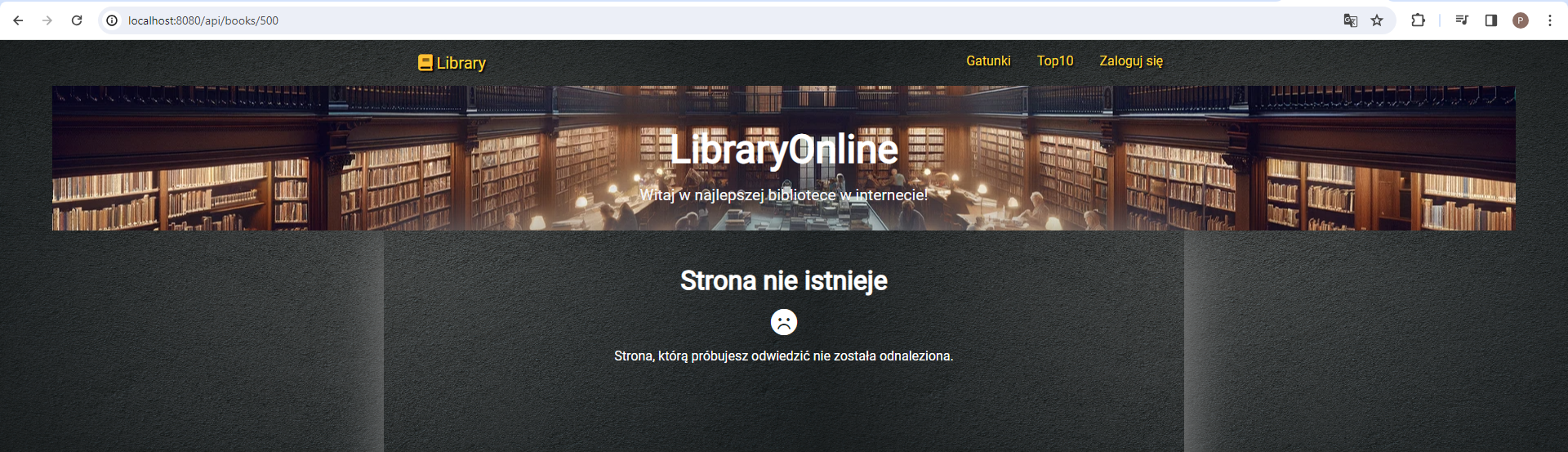


# Walidacja zapytań

Powyższe API zostały zabezpieczone przed nieprawidłowymi zapytaniami.

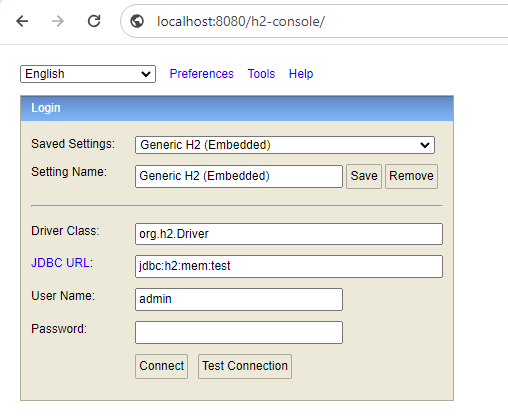


W przypadku gdy spróbujemy otrzymać dane o rekordzie, który nie istnieje, np. /api/books/500 (szukanie książki o id 500 – nie istnieje taka w bazie), to otrzymamy komunikat, że nie istnieje taka strona.



# Baza danych

Wszystkie dane, które otrzymujemy z API są zapisane w bazie danych. Aby się do niej dostać możemy dodać wejść na podstronę /h2-console.



Pola powinny być wypełnione tak jak na zrzucie ekranu. W pole password wpisać *admin*.

Po wpisaniu poprawnych danych i wciśnięciu connect, naszym oczom powinna ukazać się baza danych.

